# FITTING SURFACES TO OFFSETS IN SHIP AND YACHT HULL DESIGN: WHAT DOES AND DOES NOT WORK

Bruce Hays
George Hazen
Proteus Engineering
345 Pier One Road
Stevensville, MD 21666 USA
t+(410) 643-7496, f+(410) 643-7535
bhays@proteusengineering.com
ghazen@proteusengineering.com

## 1 Introduction

Often in the course of hull design the problem of re-creating an existing set of offsets arises. This problem can take on many different characteristics, depending on the goal of the re-creation, the origin of the offsets, and the tools available to tackle the job. All of these must be taken into account in developing the best approach to solving the problem in the most efficient manner.

It is a common assumption that there is an automatic mathematical solution to this problem, which given a set of offsets in a format familiar to naval architects, will automatically produce a fair surface which can then be used for modification, development of frames and plates, and analysis such as hydrostatics and finite element analysis. There are programs which claim and attempt to do this, and at first glance seem to succeed. *In fact, this approach almost never yields useful results*. We will discuss why this is the case, and why an approach based on manual and semi-automatic methods is very productive. We will first examine some of the issues regarding the surface mathematics being used, then describe some of the different characteristics of the basic problem, and finally discuss some successful as well as some less successful approaches to solving them.

## 2 Surface Mathematics

The process of automatic or semi-automatic fitting of a surface to offsets requires some sort of mapping of each offset point to a corresponding point on the surface. One approach that has been used is to first fit a spline to stations, thus mapping the offset points to points on the splines, and then to pass longitudinal splines through these station-wise splines to define the hull. Often this method uses interpolating splines in either the station-wise, longitudinal, or both directions (interpolating splines are splines in which the curve passes through the control vertices). In this case, each offset point on a station would be a control vertex, to guarantee that the curve would pass through it. The problem arises in the oscillatory nature of these splines, which means that although the curve will pass through the offset points, it may oscillate between them. Even if the stations are fit with approximating splines, the same problem can arise in the longitudinal fairing, which may contain oscillations. Also, this process tends to fair across any features

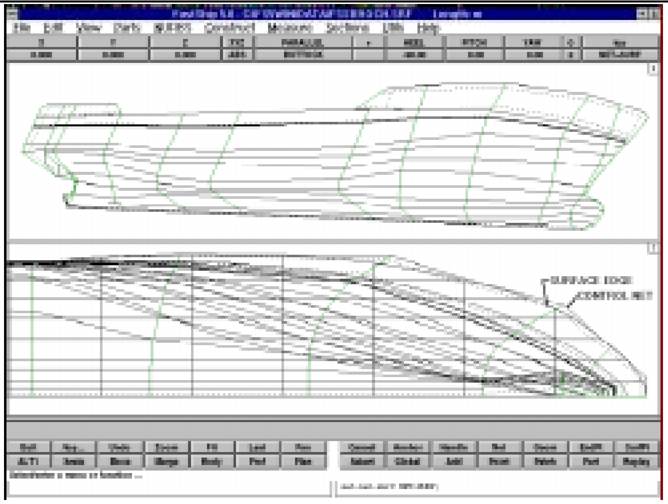

**Figure 1: Typical Ship Hull Control Net and Surface**

such as chines or knuckles, losing these features in the process. In summary, if interpolating splines are used, and control points are placed on the offset data points, the match to the offsets might be good, but the surface may well oscillate in a non-fair[1] fashion in-between the offsets. Put in terms of the problem at hand, *matching the offsets of stations is not enough if the stations and the surface between the stations is not fair*. Approaches that rely on interpolating splines suffer from this problem; it may be relatively simple to match the offsets on frames, but achieving the offsets together with a satisfactory overall shape can be considerably more difficult. Also, defining the splines, and hence the cross-fairing longitudinals, on offsets that are organized in a pattern that does not define or describe the topology of the hull will ultimately fail, because either the features will be lost, or the resulting surface definition will be overly complex.

This discussion focuses on the use of a NURBS-based hullform design program, where the surface is defined by a "control net", together with a knot vector. A NURBS control net is a grid of vertices, arranged in a series of rows and columns that explicitly define the surface shape. In Figure 1, a typical ship hull control net is shown in the upper half of the screen, consisting of a main surface with 14 columns and 15 rows (three rows are coincident at the knuckle near the main deck), and two smaller bulwark surfaces. The main surface therefore has 210 control vertices. The lower half of the screen shows a planview of the forward portion of the net and the resulting surface.

NURBS surfaces have two important properties that directly impact the task of offset re-creation. First, they are "approximating" spline surfaces, which means that the surface does not, in general, pass through the control vertices (the surface and control vertices usually do coincide in the corners of the control net). This can be seen in the lower half of Figure 1, where the planview of the control net and surface do not coincide. This means that simply placing control vertices on the offset locations will not result in a surface that matches the offsets. However, this same property avoids the tendency of oscillation that interpolating splines display. The use of NURBS surfaces can make satisfying both requirements more tractable.

The second important feature of NURBS surfaces is that of "local region of influence". This means that any control vertex does not influence the entire surface (except in very simple surfaces, which would not have enough control to realistically define a hullform). With cubic surfaces, which are most commonly used for hull design, each vertex influences the surface two vertices away from itself[2]. Beyond that point, the vertex has no influence (with interpolating splines, this is not the case, and modifications made to the bow of a hull can change the shape of the stern). This feature has the advantage that any change made to the surface is somewhat localized, and is faired in nicely over a region of the surface. However, if the control net is made too dense, the area over which a change to the control net is faired in can become too small, and local unfairness may result. Therefore it is advantageous to work with as simple a control net as possible.

---

[1] The term fair is used here in a naval architectural sense, rather than a mathematical sense. For example, a mathematician would call an oscillating waterline fair as long as it did not have any sharp changes in slope or curvature, while a naval architect would not.

[2] Cubics most closely match the way in which objects bend in nature. Also, cubics are the lowest order which guarantee that the surface will be both slope and curvature continuous from one interval to the next. Higher and lower orders can be less intuitive in the editing process as well.

# 3 Description of the Problem

## *3.1 The Goal*

The first question that must be asked is what the surface that is being fit to the existing offsets will be used for when it is completed. Some of the possible uses for the surface, and their requirements are shown in Table 1.

**Table 1: Surface Uses and Requirements**

| Use of the Surface | Requirements |
|---|---|
| Construction (lines, frame and plate definition). | The final surface must be extremely fair. Therefore, if the surface matches the offsets exactly, the offsets must be extremely fair to begin with. Otherwise, the surface should be faired knowing that the offsets will not be exactly matched (also known as lofting). |
| Computing intermediate sections, for example for use in other programs such as finite element analysis or hydrostatics, where fairness is not a high priority. | The surface can be a close or exact match to existing offsets, without worrying about how many vertices are used in the control net to describe the surface, since the surface will not be modified, and fairness is not critical. |
| Design, where the hullform may change as the design progresses. | Fairness of the final surface is more important here than an exact match of the offsets, since the hull may change anyway. Also, the control net must not be too dense, so as to be manageable for further modification. |

Two important questions arise from these considerations:

1. **Should the offsets be exactly matched or should they be faired?** In the next section we will discuss the origin of the offsets, and their assumed quality.

2. **How dense a control net can be tolerated?** If the surface is to be used only for interrogation (cutting sections, measuring hydrostatics, measuring surface properties, lofting plates and frames), and will *not* be modified further, a net that is too dense for manual modification may be acceptable. Otherwise, the control net that is created must be reasonable in the quantity and placement of the control vertices.

## *3.2 The Offsets*

Knowing the source and quality of the offsets to be matched is very important, as well as having access to the original source to check offsets and to get more data.

### Offset Quality

The quality of the offsets that are being matched is very important in the process of re-creating the hull. Are the offsets measured from an existing hull or from a lofted book of offsets? If so, it might not be unreasonable to try to match them very closely. On the other hand, if they are measured from a scaled lines drawing, they are not likely to be entirely fair or accurate. In this case, you probably do not want to match them exactly, but rather want to design a fair surface that is "somewhat" close to the offsets. In other words, fairness of the surface is the overriding requirement.

In many cases, a book or file of offsets will contain erroneous data. This data must be either removed or corrected so that the error does not influence the final surface shape. Having access to the original set of lines can be very important in this case.

### Offset Format

Often the offsets being used in the matching process are only frames or stations. Stations can do an adequate job of describing the shape of a hull in the midship area, where the waterlines are mostly parallel to the centerline. But in the ends of the vessel, where the shape is changing rapidly as one moves forward or aft, stations are not the best way to describe the shape[3]. Stem and stern profile shapes are especially poorly defined using only stations. If possible, the offsets that you work with should include the centerline profile, waterline endings at both ends of the vessel, and offsets of any chine or knuckle in the hull, as well as feature lines such as the flat-of-side and flat-of-bottom. The human eye is remarkably adept at seeing knuckles and feature lines in station offsets, even when they are not defined explicitly by curves. The computer is just as remarkably poor at finding these features when they are not defined explicitly.

Time spent preparing the offsets to be used in the re-creation process is time well spent. Make sure that the data is error-free, and that the vessel is well defined by the offsets, especially if it is critical that the offsets be matched closely. Your trained naval architect's eye may be able to easily interpolate the shape of the hull from a small number of offsets, but the computer will not be able to do so.

# 4 The Problem with an Automatic Approach to Surface Fitting

## *4.1 Overview*

Many programs and algorithms exist for "skinning" a surface through a series of offsets, including most major CAD systems, such as ProEngineer™, Autodesk's AutoSurf®, Intergraph EMS™, as well as some hull design programs. Why not just push the button in one of these programs and call it a day? After all, they claim to create a surface that passes through the given offset points, and NURBS mathematics tells us that a surface can be found that will pass through each of the offset points. This is a classic case of "when things seem too good to be true, they usually are".

The first problem with this approach is that it assumes that the offsets completely define the hull, and that they are perfectly fair (a rare case indeed). Even if the offsets are fair and very complete, they are usually organized in a station-wise fashion, which is not consistent with the manner in which points on a surface are distributed, especially in the ends of the hull. This can cause surfaces that are unnaturally contorted, as in Figure 2.

The second problem is that the surface that results from such a process is usually far too dense to allow any reasonable modification by hand (in the extreme, a direct solution of a surface to match a set of offsets creates a control vertex for every offset point). So if you want to change the surface because it suffers from the use of incomplete, inaccurate, or unfair offsets, it is impossible to do so in a practical manner.

---

[3] This is why hydrostatics programs that operate on stations require many stations closely spaced at either end of the vessel to give accurate results.

Finally, as mentioned previously, it is very difficult for automatic methods to find topological features such as chines, tangent lines, etc. These important features of a hullform are usually lost during a fully automatic re-creation process.

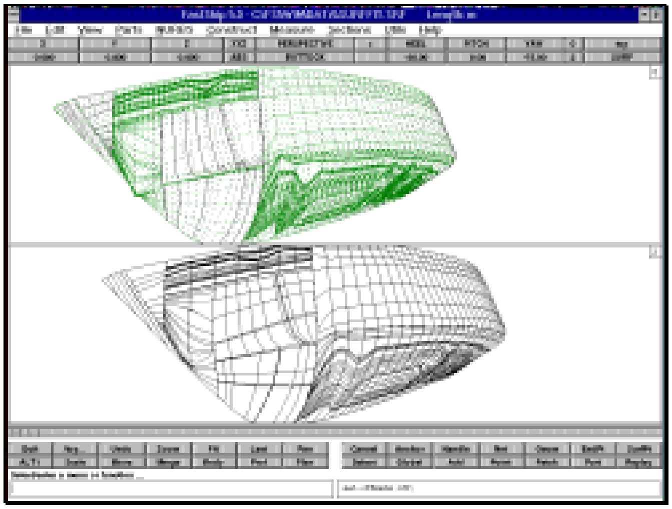

**Figure 2: Surface Resulting from Automatic Fitting**

Figure 2 shows a surface that was created by automatically skinning a set of offsets, on a relatively simple hull with no distinct features such as chines. The upper half of the screen shows the resulting control net, with 4850 vertices (a similar hull designed from scratch might have 100 vertices). The lower half of the screen shows the surface that results. The figure shows the overwhelming density of the control net, the fundamental unfairness of the surface, and even a control vertex on the wrong side of centerline. Even with all of that, the lower half of Figure 3, which shows sections cut through this surface, is striking in how reasonable it looks at first glance, and how close the fit is to the original offsets. However, close inspection of these sections reveals that they are not fair and could not be used in construction. This dense control net could be used for interpolation of more frames for use in finite element analysis or hydrostatics, where fairness and accuracy are not critical, but is far too dense to allow for practical manual modification.

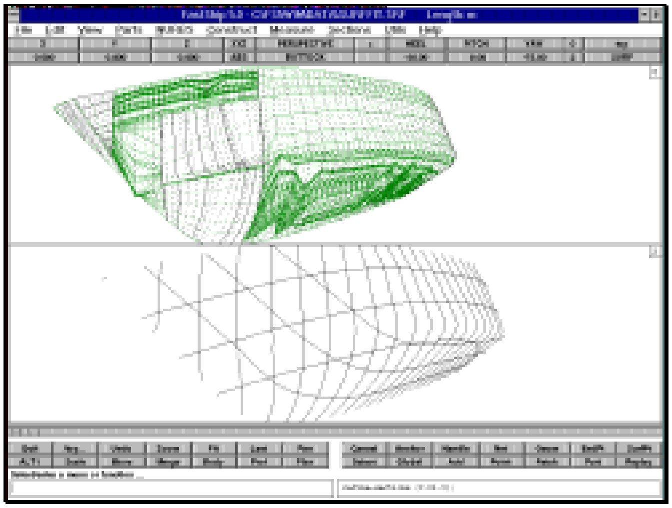

**Figure 3: Sections Resulting from Automatic Fitting**

### *4.2 Curve and Surface Fitting*

Why do these automatic methods fail? In order to better appreciate the difficult issues involved in surface fitting, it is useful to compare and contrast this activity with the more common and well understood process of fitting analytic curves to point data.

It is commonly assumed that the only input required to successfully fit point data with a curve is the set of points themselves. Perhaps such a notion is a holdover from graphing exercises in school when the student was given ordered pairs of numbers to plot. What is readily overlooked is the implicit assumption that these pairs of numbers are presented in a meaningful sequence thus indicating the connectivity between points. Once the curve fitting exercise is reduced to a numerical procedure this implicit sequencing is typically codified as a monotonically increasing parameter.[4] Here the curve fitting practitioner starts making assumptions that are not explicitly contained in the data he has been given to fit. At a minimum, he must answer the following two questions:

- What sort of function should be used to interpolate or approximate the data?
- How should parametric values be assigned to the individual data points?

Polynomials and piecewise polynomials (splines) are frequently chosen candidate functions for fitting because of their versatility, well know properties and computational efficiency. In special cases, where there is a foreknowledge of an expected functional relationship between the points to be fit, the appropriate functional form can be used. Though picking an appropriate functional representation often receives the greatest amount of consideration, frequently it is the assignment of parametric values that determines the ultimate success or failure of the fitting exercise.

Mathematically our only requirement for this assignment is that our chosen parametric values increase monotonically as we travel from point to point in the desired sequence. Typical choices for this parametric assignment are based upon sequence number or spatial distances between sequential points.

---

[4] Here we will only consider parametric curves and surfaces as many naval architectural surfaces and sections cut through them are multi-valued. Explicit functions in terms of the Cartesian coordinates are unsatisfactory for general application to such surfaces and curves.

For nearly evenly spaced data sets there is little discernible difference between the geometric representation of curve fits using either of these parameterizations, but when the spacing is irregular considerable differences result. It is up to the practitioner to decide whether or not the resulting curve is an appropriate functional representation of the data. This subjective appraisal is true whether or not the point data was interpolated or approximated — this is the hidden reality of fitting functions to point data; someone has to be responsible for deciding whether the geometric properties of the resulting function represent the implicit character and form desired!

The ambiguities to be resolved when curve fitting data seem minor when compared with these same considerations as manifest when fitting data to functional surfaces. This is especially true when those surfaces are intended to be general enough to capture all of the geometric diversity found in naval architecture. In order to offer concrete examples of both the problems and strategies to overcome them, we will limit the following discussion to surfaces with two independent parametric variables.[5]

Once again, the surface fitting practitioner must choose an appropriate class of function for the job. In the case of a tensor product NURBS surface, this means choosing the degree and the number of defining control points for the splines in each independent direction. Though the degree and number of spans in each parameter need not be the same, the grid of control points must be rectangular; i.e. each control point can be thought of as belonging to a particular row and column. This observation leads directly to the realization that surface fitting is nearly always a task of approximation rather than interpolation, since interpolation requires that the input data also be rectangular. Certainly ship offset tables seldom meet this requirement, so a least-squares approximation to the input data is formulated instead.

When considering a NURBS surface approximation to a point data set, our intrepid surface fitter must once again begin by assigning parametric pairs to each point in that data set.[6] Here simple algorithms can rarely be formulated which are generally applicable. Attempts to use girth based parametric assignments typically fail unless all of the data points can be organized into rows or columns. To escape the requirement for a rectangular input matrix of data points, the rows or columns can be curve fit first and then each resulting curve can be evaluated at a consistent number of points along its length. In naval architectural practice, this typically involves fairing transverse sections and then interpolating longitudinally. Notice that this cross-fairing approach requires that each curve be assigned an appropriate parametric value. Though the practice of cross-fairing between stations and iso-girths has had many practitioners, it has enjoyed limited success, frequently requires exhaustive data preparation and utterly fails when confronted with all but the most straightforward hullforms.

Once again the success or failure of our attempts to fit functional surfaces to point data is not so much determined by our mathematical methods, but rather by our ability to capture the essential topology of the surface. Fitting common geometric features including chines, bow and stern bulbs, and embedded surface flats all requires a heuristic understanding of naval architectural practices and purpose. Such an adaptive topological understanding is beyond the ken of automatic fitting algorithms, but not the naval designer.

---

[5] This category of surfaces covers both tensor product surfaces (typically NURBS) and transfinite surfaces such as Coons patches. It does not include surfaces based on triangular patches that are described in terms of three barycentric coordinates.

[6] For NURBS there are two interdependent activities, assigning values to the basis knot vectors and assigning pairs of parametric values to each Cartesian data point. Both of these assignments will impact the computed optimal distributions of control points and will thus impact the detailed geometry of the resulting surface.

Thus, the most effective surface fitting can be achieved by including the naval architect/surface designer in the process. With a ‘candidate’ surface provided by the designer that includes all relevant topological features, the data set can be projected onto this surface where the implicit parameters can be evaluated and assigned to each data point. Now that each point in the data set has both Cartesian and parametric coordinates, a least-squares fit solution can be computed for the surface control points. Such a scheme allows and encourages the designer to further modify each successive candidate surface and then repeat the fitting process until a satisfactory surface model is produced. This process, which can be considered to be semi-automatic surface fitting, will be discussed further in the next Section.

## 5 The Manual and Semi-Automatic Approaches to Fitting

There is a better, more practical approach to the re-creation of offsets. With ***FastShip***, we have provided a number of tools to make the process as easy and efficient as possible, while creating useful, practical results. These tools can be divided into two categories: manual fitting tools, and semi-automatic fitting tools. Although these tools are not an automatic, instant solution to the problem, they represent significant capability to realistically and efficiently solve the problem of re-creating an existing set of offsets.

### *5.1 Manual Fitting/Fairing*

The basic approach to re-creating a set of offsets is to manually create a surface, and compare it to the target offsets, and refine it as necessary until a satisfactory match is achieved. With this approach, the designer can build a surface that has the correct topology (chines, knuckles, circular bilge turns, conical bow, straight sections, parallel midbody, etc.), without relying on an automatic program to discern all of these features from the offsets. An important part of this process that it gives the designer the ability to make the tradeoff between matching the offsets and assuring fairness, selectively ignoring offsets when appropriate. This is why this process is referred to as both “fitting” and “fairing”.

***FastShip*** supplies a number of important tools to help with this process:

- **Display of offset data and/or markers superimposed on surface**

Any offsets can be displayed on the screen superimposed on the surface, and sections of the surface being designed at the same locations as the target offsets. This gives immediate display of the differences between the offset file and the surface being designed.

- **Real-time editing**

To help speed the process of modifying the surface to match the offsets, ***FastShip*** displays the sections rubberbanding in real time as the control vertices are moved. With this capability and the display of offsets, the designer can interactively position the control net vertices to give the best match to the offsets.

- **Global dragging with multiple viewports**

Interactive editing is further enhanced when the surface and offsets can be viewed in multiple viewports, to show different views of the hull at the same time. With Global Dragging, the real time rubberbanding is displayed in all viewports.

- **Measurement tools**

***FastShip*** allows the designer to measure the absolute location of any point on the surface, as well as measuring the distance between an offset and a point on the surface. Also, hydrostatics can be computed almost instantly, to give information on whether the surface has the intended volumetric properties. Fairness can be measured by surface curvature and curvature along any section.

- **Double-net function**

There are times when more control vertices are needed in a portion of the surface in order to give enough control to match the offsets in that area. At any time in the design process, a new “row” or “column” may be added into the control net. However, this has the disadvantage of changing the shape of the surface. To solve this problem, ***FastShip*** has a function that can double the number of rows or columns (or both) in the surface, without changing the surface shape. While this adds to the complexity of the control net, it also greatly increases the amount of local control. This function can be used together with the *split-net* function, which allows a portion of the surface to be split off as a separate surface. The double-net function can then be used on just the portion of the surface that has been split, so that the remaining portion of the control net does not become more complex.

- **Macro to report offset differences**

  ***FastShip*** provides a macro to print a table of offsets, together with the “error” between the surface and the target offsets, including a least-squares summary. Also, an error-band summary is given, tallying the percentage of offsets within given error tolerances. This error-band approach is sometimes used as the specification for how close a given set of offsets must be matched.

In its most basic form, this manual process begins with a “flat sheet” of NURBS surface, which is then stretched and molded to create the desired shape, with additional control vertices being added as necessary as the design process continues. The time spent in reproducing a hull can be significantly reduced if an existing surface in the designer’s library has a similar topology to that of the offsets being re-created. For example, most container ships are topologically similar; they are single screw, have a flat-of-side and -bottom, circular bilge turn, bulbous bow, and bulwarks. Even if an existing surface is not close to the same size as the target offsets, if it has the correct topology, it can be quickly scaled to a new size, and the surface can then be locally modified to match the target offsets.

## 5.2 Semi-automatic Fitting

***FastShip*** includes a function (“*fit-surface*”) to automatically modify the control net so that the surface matches the offsets in the best least-squares sense. With this function, the designer first creates a surface that is close in shape to the offsets, with the necessary topological features (chines, etc.). This function works quite well in areas of the surface that do not contain discontinuities, such as tangent lines and chines. However, it can produce unpredictable results near the edges of surfaces, so the designer should manually fit the edges of the surface and any interior feature curves such as tangent lines and chines. Then these areas of the surface may be excluded from the *fit-surface* process with the *freeze-net* function before proceeding with the *fit-surface* function.

The *fit-surface* function does not add or delete rows or columns in the net; it only moves existing control vertices. This maintains the density of the surface, so that an overly complex surface is not created (as was the case in Figures 2 and 3). It can be selectively applied to only portions of the surface, so that if one area needs to have a very close fit, while another area does not, it can be used only on the appropriate portion of the surface.

The fit-surface command is most effectively used in an iterative process, alternating between the designer’s fairing and the computer’s fitting. Fairness is therefore still in the hands of the user, with the usual tools for measuring fairness (curvature measurement along curves, surface curvature measurements, etc.). If the result of a *fit-surface* command fits the offsets well, but is not fair in the designer’s opinion, the operation can be reversed with the *Undo* command. An even better approach is to modify the new surface by hand with fairness as the constraint, and then to run the *fit-surface* command again (the *fit-surface* command can take between 3-20 seconds, depending on the speed of the computer, the density of the offsets, and the complexity of the surface). This process can continue as long as is necessary to successfully make the tradeoff between the matching of the offsets and the fairness of the surface.

## 6 Summary

The creation of a surface to match a set of offsets is a complex problem with many different facets. It is not really a single problem, but many different ones, each with its own set of inputs, constraints, and desired results. Rarely do these combine to create a situation where a totally automatic solution is satisfactory.

Spline-based solutions have a tendency to create over-defined models, which can contain oscillations that are difficult to remove even manually. They also can fair over and lose features in the surface, and for these reasons, they are an unsatisfactory solution in an automatic, semi-automatic, or manual mode.

NURBS surfaces, when used in a fully automatic approach, can also suffer from over-defined models, and the loss of key topological features in the surface. The best solution to the re-creation of offsets is a semi-automatic method using NURBS surfaces, where the designer creates a surface with the correct topology, which the computer then modifies to create a best fit to the offsets.

It is important that the designer understand the source and quality of the given offsets, to know whether it is important or even possible to exactly match them (why match a set of offsets exactly if they have not been faired already?). If the surface is to be used only to compute sections to be used for hydrostatics calculations, fairness is a low priority, and accuracy is not too important as long as the integrated properties (volume, center of buoyancy, etc.) are matched.

Usually, fairness is the overriding requirement in the process, which means that the target offsets may not be matched exactly. In this case, manual and semi-automatic approaches are the most productive, producing surfaces that can be faired reliably and modified easily.